RasPi

DESIGN
BUILD
CODE

26

Get hands-on with your Raspberry Pi

HACK YOUR TV

+

SET UP A WORDPRESS WEBSITE

Plus Turn your Pi into a stop-motion studio

Welcome

TV isn't what it used to be. I'm old enough to remember when the UK got its fourth (!) channel, and when remote controls still connected to the set with physical wires. Upgrading to an infrared remote was something very special. Many kids took great delight in surreptitiously pressing the remote's buttons while their folks were watching the Six O'Clock News, and in the chaos that ensued. In the days of Netflix and Amazon Prime our old infrared remotes gather more dust than they used to, but where there is infrared there is a Pi that can hack it. Now you can reboot one of the most fun pranks of the Eighties, improved with today's technology. Kind of like *Stranger Things*, but different...

April

Editor

Get inspired

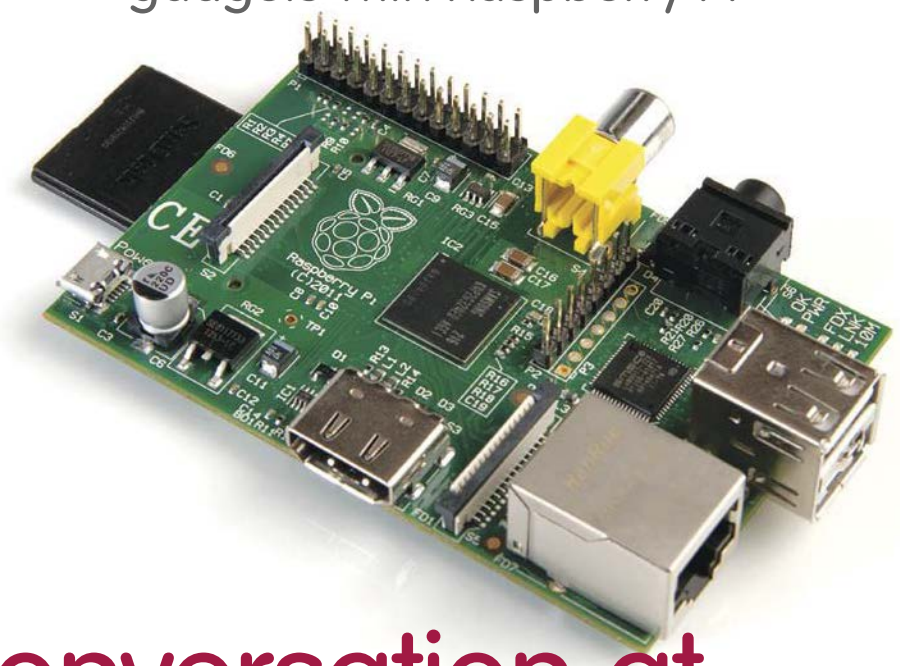
Discover the RasPi community's best projects

Expert advice

Got a question? Get in touch and we'll give you a hand

Easy-to-follow guides

Learn to make and code gadgets with Raspberry Pi

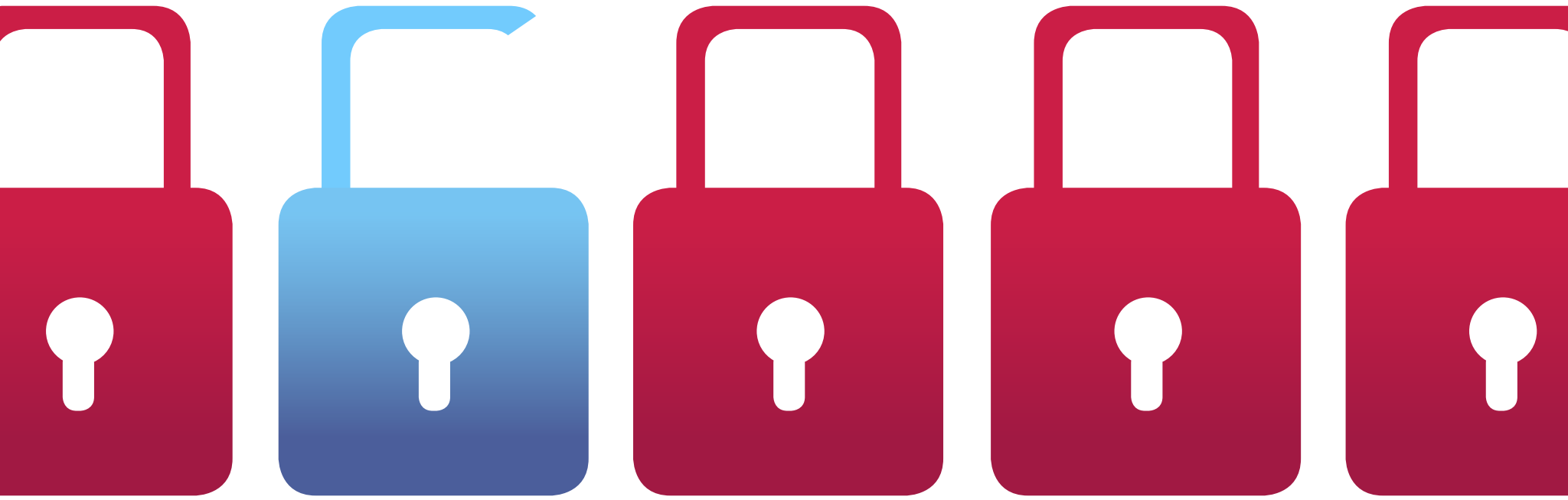


From the makers of

LinuxUser & Developer

Join the conversation at...

@linuxusermag | Linux User & Developer | RasPi@imagine-publishing.co.uk



Contents

Hack your TV with Pi

Use the Energenie IR board and a Pi as a TV remote control

The Cold Boy

A fridge/GameBoy combo? Yes please

Set up the official 7-inch Pi Display

Assemble this display module and get it up and running with your Pi

Set up a WordPress website on your Pi

A quick guide to running a WordPress server

Turn your Pi into a stop-motion studio

Build your own animation studio using your Raspberry Pi

Learn about your Pi

Look at how you can monitor the Pi to see what is happening

Talking Pi

Your questions answered and your opinions shared

Hack your TV with Pi

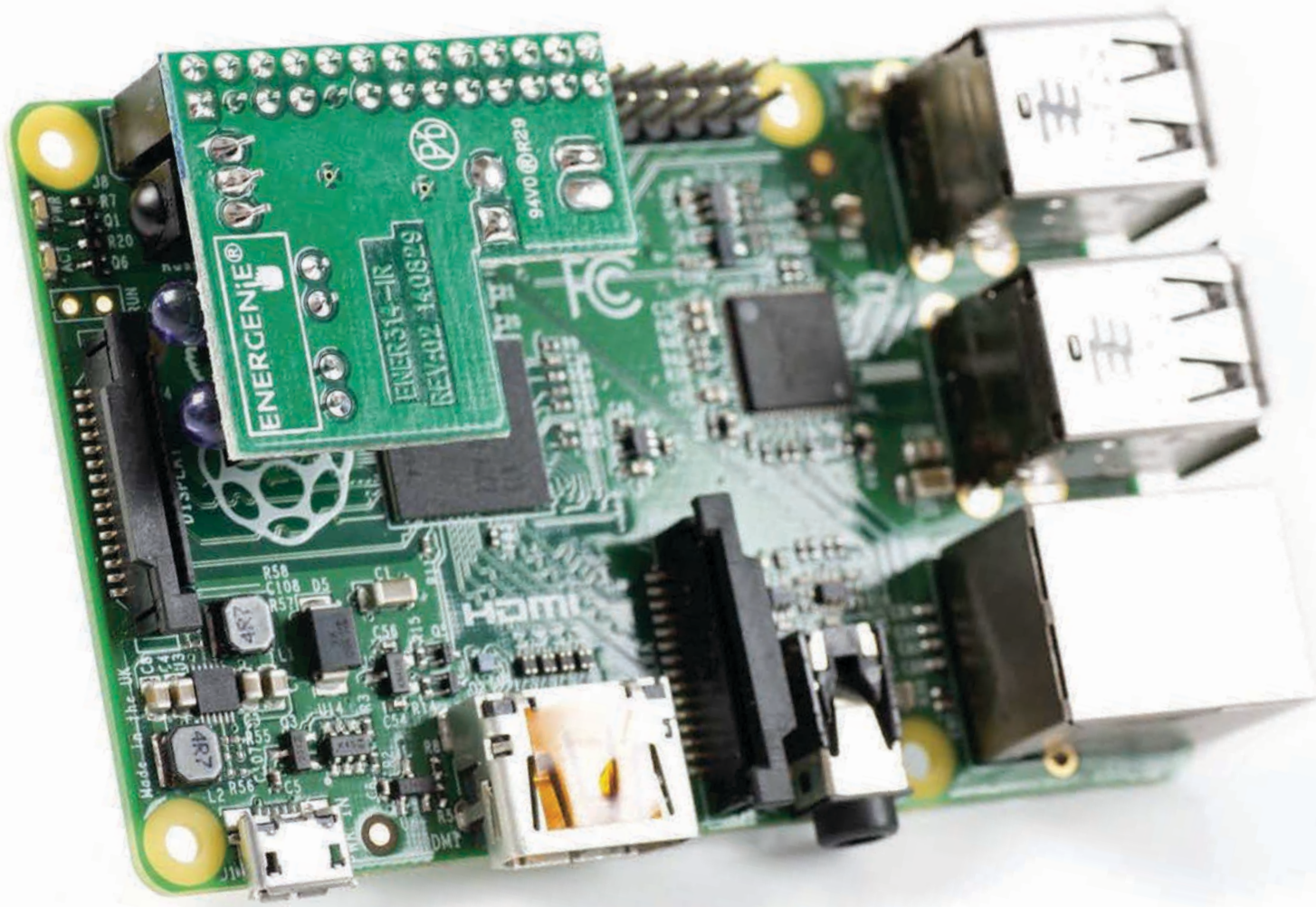
Use the Energenie IR board and a Raspberry Pi as a remote control for your television

In this project you will learn how to emulate your television remote using your Raspberry Pi and an Energenie IR board in order to control your big screen. Why? So you can change the channel while no one is looking! Infrared, or IR for short, is light with a wavelength greater than the red end of the visible light spectrum, but less than that of microwaves. Infrared radiation can't be seen with the naked eye, but it can be felt as heat energy. Infrared radiation is used to transmit data from device to device, including between remote controls and their televisions, Blu-Ray players or to provide data links over short distances between computers or mobile phones.

This tutorial will show you how to set up the Energenie's Pi-Mote IR board, which will enable your Raspberry Pi to learn infrared remote-control signals and then transmit these same commands in order to control your device. This tutorial will focus on controlling a television in particular, but remember that the board is compatible with a wide range of devices. This means you can turn your Raspberry Pi into a remote for a range of household objects, such as your media system, smartphone dock or even the air conditioning.

THE PROJECT ESSENTIALS

Energenie IR module board
http://bit.ly/1MdpFOU



01 Edit the config.txt file

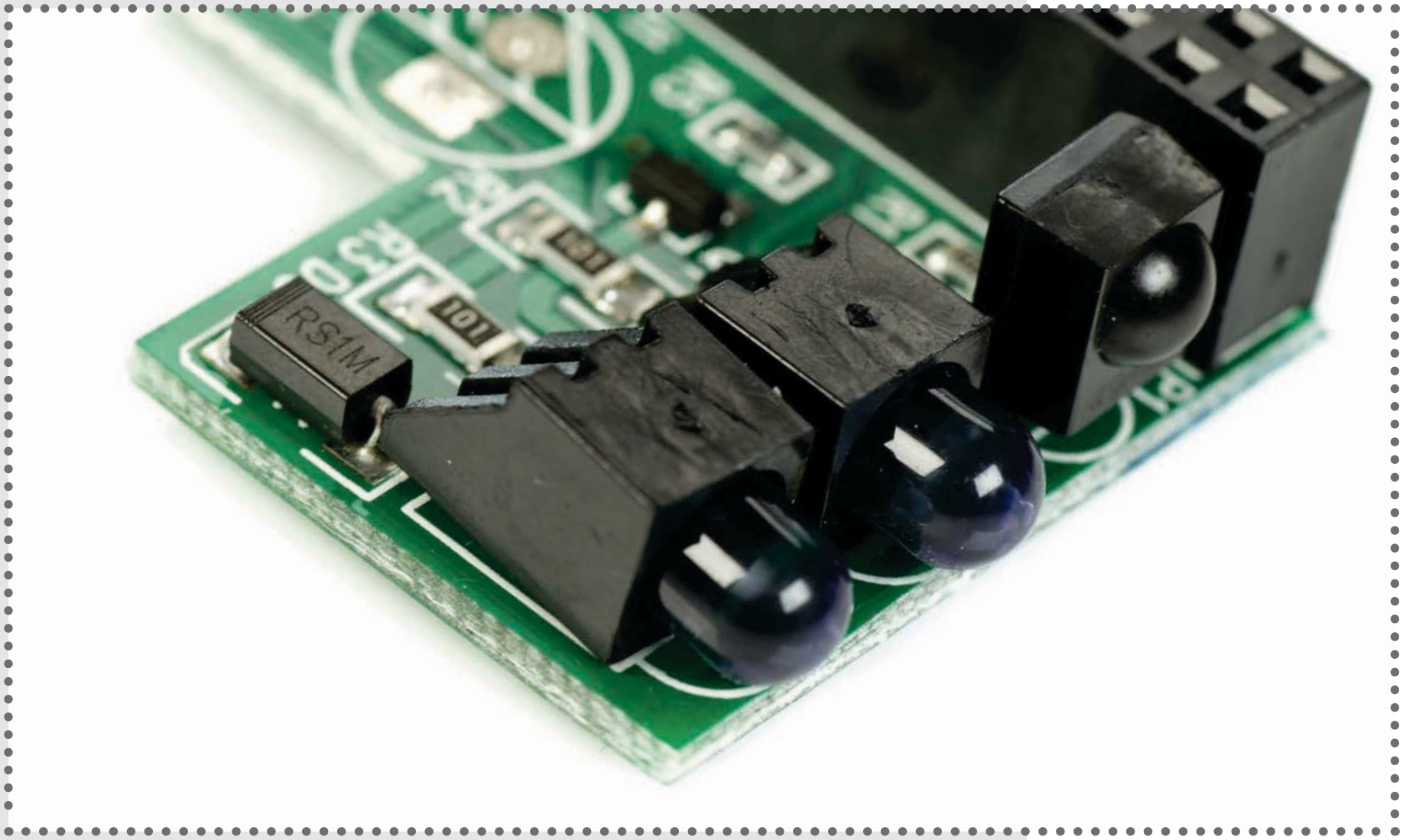
Attach your IR board module to your Raspberry Pi and boot it up. It is always advisable to update your SD card software. In the LXTerminal type:

```
sudo apt-get update
sudo apt-get upgrade
```

Next add code to the /boot/config.txt file to enable the LIRC IR software and IR module to interact. In the LXTerminal type:

```
sudo nano /boot/config.txt
```

This will load the config .txt file. Scroll to the bottom of the text and add the following line:



Above Signals are sent from the Pi-Mote to your TV through the board's IR lamp

```
dtoverlay=lirc-rpi-overlay
```

Now press Ctrl+X and save the file, then reboot your Raspberry Pi by typing sudo reboot.

02 Install the software

Next install the LIRC software; this stands for Linux Infrared Remote Control. It is the program that enables you to interact with your television and transmit commands accordingly. In the LXTerminal type:

```
sudo apt-get install lirc
sudo apt-get install lirc-x
```

Restart your Pi. The default GPIO pins used when no pins are specified are pin 12/GPIO 18 for input for

received infrared signals and pin 11/GPIO 17 for output for transmitted infrared signals.

03 What are all these files?

It's worth clarifying some file names and folders. All of the files used are stored in **/etc/lirc**, and there are two main ones: the hardware and LIRC configuration files (**hardware.conf** and **lircd.conf**). The latter holds all the data about your remote control, such as signal length, names of buttons and header details. This is the file to edit to emulate a remote control.

04 Test the IR receiver is working

To test that the IR receiver will 'pick up' the transmission from your remote, you need to stop the LIRC daemon, enable the test mode and then start the mode 2 testing. This runs a program to output the mark-space of the IR signal. It measures the pulse and space length of signals, returning the values to the terminal. Open LXTerminal and enter the commands below, then grab your control, point it at the IR receiver and press some buttons:

```
sudo /etc/init.d/lirc stop
sudo modprobe lirc_rpi
sudo mode2 -d /dev/lirc0
```

You should see something like this:

```
space 16300
pulse 95
space 28794
```

Lircd daemon

The LIRC enables you to decode and transmit infrared signals. The one used in this project is the lircd daemon that decodes IR signals received by the device drivers and accepts commands for IR signals to be sent if the hardware supports this. You could adapt your project to create a remote control for your media device using a simple Apple remote control as the input.

```
pulse 80
space 19395
space 28794
pulse 80
```

05 Transmit a signal

Now your remote is recognised and the IR board is working, you can use the **irsend** tool to record the signals and use these measurements to send commands to your TV. First alter the hardware.conf file located in the /etc/lirc folder:

```
sudo nano /etc/lirc/hardware.conf
```

Make the following changes:

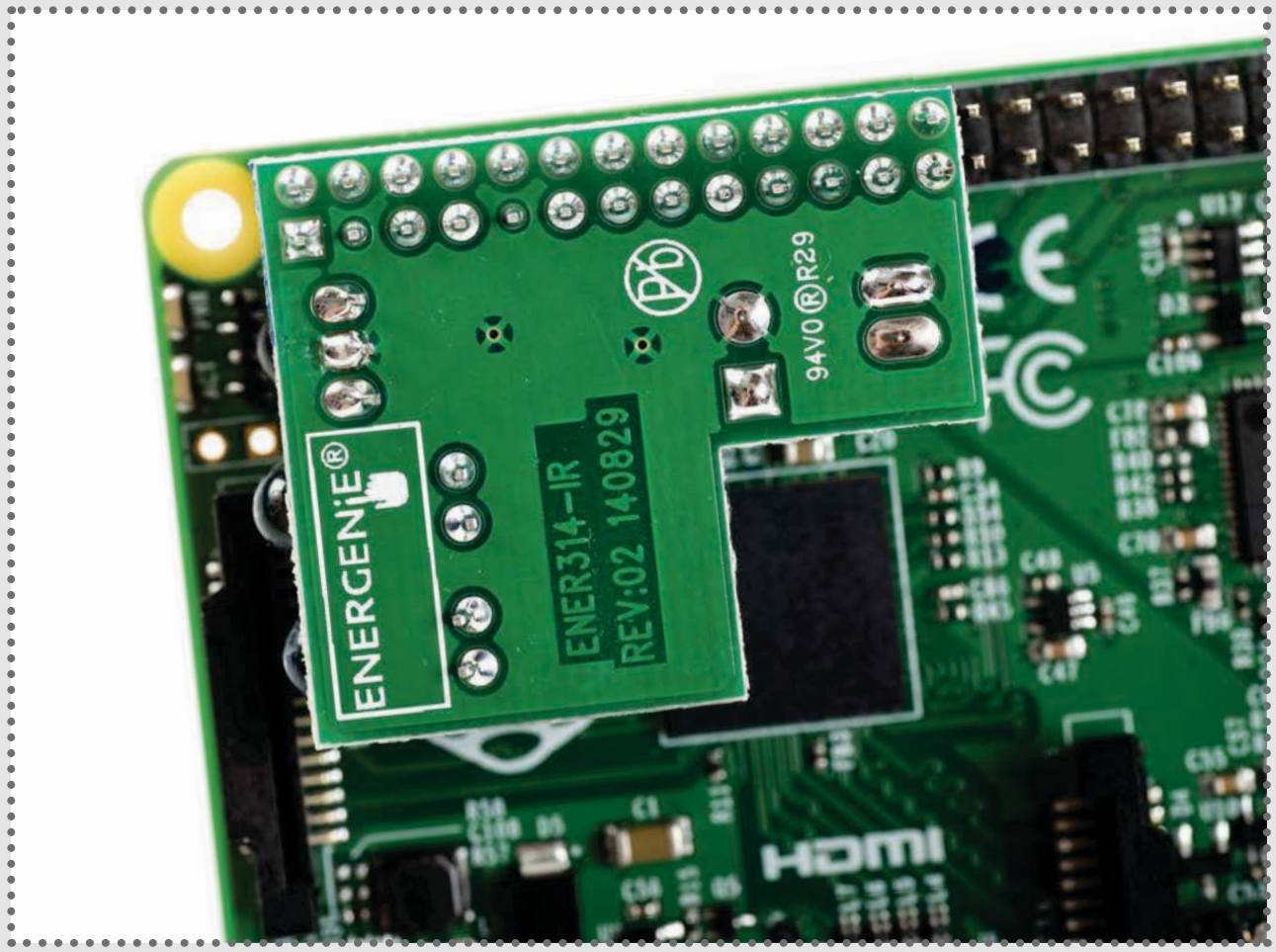
```
LIRCD_ARGS = “--uinput”
DRIVER = “default”
DEVICE = “/dev/lirc0”
MODULES = “lirc_rpi”
```

Press Ctrl+X to save the file – but don’t rename it – then press Y and Return. Now restart the lirc daemon by typing :

```
sudo /etc/init.d/lirc restart
```

06 Get a new LIRC file

Next locate a compatible lircd.conf file that contains all the information you need to know about the remote’s buttons and each of their specific functions (Instead of buttons they are referred to as keys). The good news here is that you do not need to create this file from scratch, because it is possible to use or adapt an existing remote control configuration file. These can be found on



Left Energenie's Pi-Mote controller board costs £10, and you can get RC plug sockets with it for an extra £10

LIRC supported remote index websites, such as http://lirc.sourceforge.net/remotes. (Some remote configuration files will not be available on the LIRC index. This is because not all remotes are supported by LIRC. You may find more up-to-date files hosted on the GitHub website.) Download the correct file for your remote and save it in the Pi/Home folder. Note that we're using a Samsung remote in this tutorial.)

07 Use your new LIRC file

Once you have a suitable LIRC file, copy and paste over the code from the file that you have found into lircd.conf, which is stored in the /etc/lirc folder on your Pi. This will overwrite any current configuration setup that you have, but it is only an issue if you have already set up a configuration file. Open the existing lircd.conf file by typing:

```
sudo nano /etc/lirc/lircd.conf
```

Now copy and paste over the code from the new LIRC

file. Press Ctrl+X to save the file, but do not change the name. Then restart the LIRC by typing **sudo /etc/init.d/lirc restart**. It will say that it failed to stop the daemon, but this is because it is already stopped and therefore it cannot be stopped again!

08 Test the lirc.conf file

Assuming that the lircd.conf file is compatible and your setup is successful, you can now test that it works. In the terminal window, type **irsend LIST Samsung " "** (replacing Samsung with the name of your television remote, which is stated at the top of the lircd file). This code will list all the KEYS (buttons) that are installed in the lircd.conf, displaying a list of the commands that you can send to your television.

09 Hack your television

Now that your lircd.conf configuration file is recognised, you are ready to control your television. The line of code is very simple and follows the format:

```
irsend SEND_ONCE Remote_Name Remote_Button
```

For example, to control the TV Menu you would type in the LXTerminal: **irsend SEND_ONCE Samsung KEY_MENU**. This will send the Menu IR signal and the menu will appear on the TV. In order to send a different button signal, alter the KEY_ field. The key prefixes can be found in the lircd.conf file or by listing the keys with: **irsend LIST Samsung " "**.

10 Make your own lircd.conf file – part one

Sometimes you may not find a compatible lirc.conf file and instead you have to create one yourself. This

Create a UI

It is possible to combine the IR board, the LIRC program and a web interface, which would open up many possibilities for projects. Combining the GPIO pins with a web server means that you can create a user interface that can be used to control your devices. Change the channel from your laptop or phone, turn the volume up or down or turn the TV off. A starter project can be found at **http://bit.ly/104CaMU**.

involves running a program called irrecord, pointing your remote at the IR board and then simply pressing loads of buttons! This will then record the signals from your remote where you can assign KEYS to each of the signals. Stop the LIRC software by typing in the terminal:

```
sudo /etc/init.d/lirc stop
```

11 Make your own lircd.conf file – part two

Next create a new lircd.conf configuration file and save the output. In the LXTerminal type:

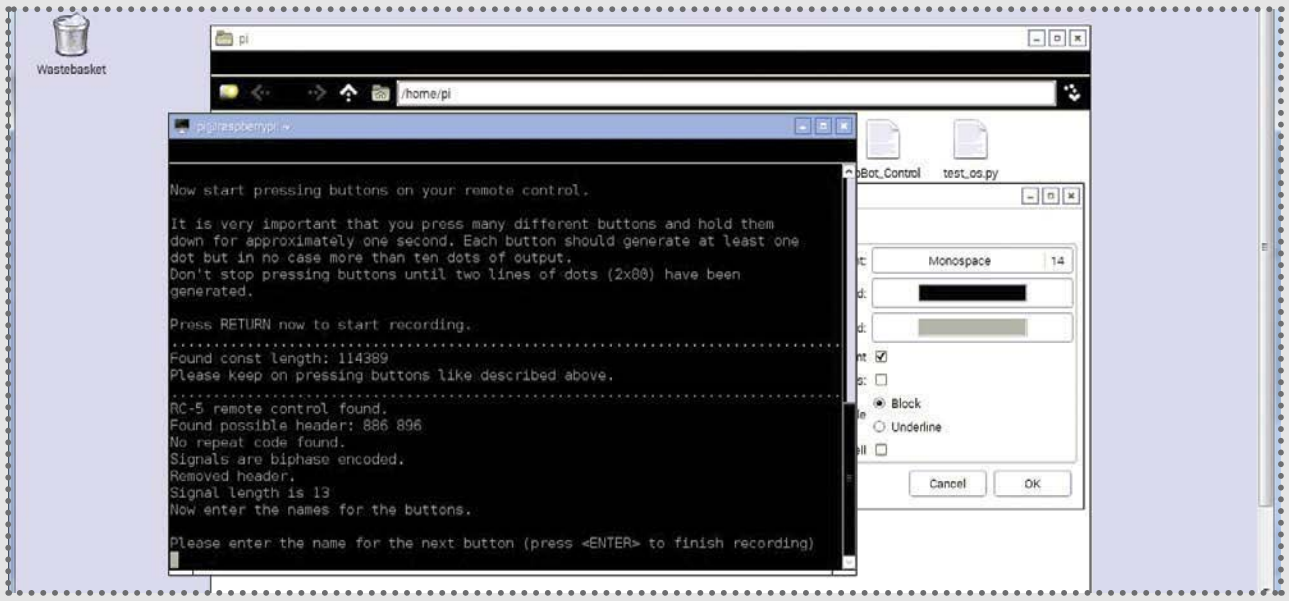
```
irrecord -d /dev/lirc0 ~/lircd.conf
```

This will open the 'create' program and will present you with instructions on how to record and save the signals. There are two stages to this: the first part involves you repeatedly pressing the buttons on the remote until there are two lines of dots on the screen. This measures and records the signals being sent from the remote. Do this in a logical order, starting at the top of the remote and working downwards.

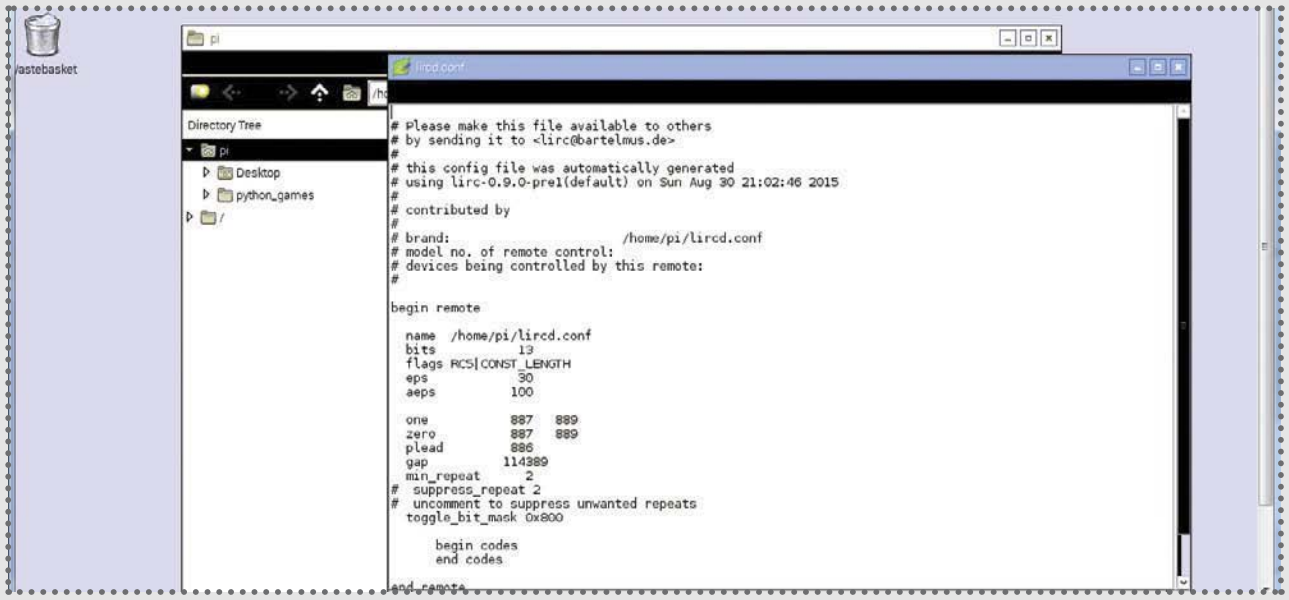
12 Make your own lircd.conf file – part three

Once the two lines of dots have been completed, your remote has now been recognised. The program will ask you to enter the names of the keys for each of the signals it has recorded. Follow each of the on-screen prompts, typing the names for each of the remote buttons/keys. For example, type KEY_UP and then press the corresponding 'up' key on the remote. You will

"Sometimes you may not find a compatible lirc.conf file and instead you have to create one yourself"

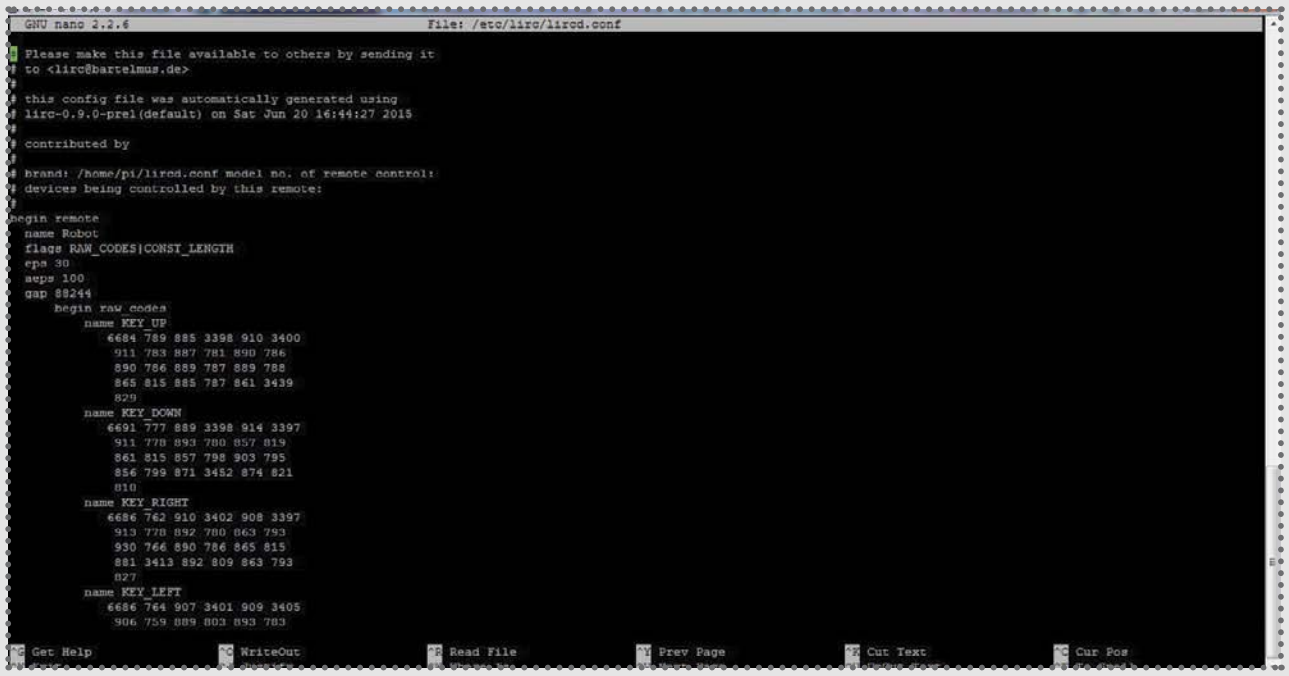


then be prompted to type in the name of the next key, for example KEY_BACK, then press the 'back' key on the remote and so on. Keep doing this until you have entered names for each of the recorded keys.



13 Rename the remote

Unlike the ready-made lircd.conf files, the name of the remote on line 14 will probably be set as "home/pi/lircd.conf". Under the heading begin remote, find the name label and rename "home/pi/lircd.conf" as something else. This makes it easier to refer to in the code line, for example, as "Samsung".



14 Transfer the lircd.conf file

Now your lircd.conf file is ready to transfer to the /etc /lirc folder, as shown previously in Step 7. The simplest method is to copy and paste over the code that you have just created, but this will overwrite any old configuration file setup that you have. If you want to keep a previous configuration then follow Step 15, else jump to Step 16. In the LXTerminal, type:

```
sudo nano /etc/lirc/lircd.conf
```

15 Not overwriting

If you have already set up a lircd.conf file or you want to use a new one and still keep the old one, then you can create a new configuration file. This is automatically saved in the /home /pi folder and can be copied over to the /etc/lirc folder. Firstly, make a backup of the original lircd.conf file, creating a copy of the file and saving it as lircd_orginal.conf. In the LXTerminal, type:

```
sudo /etc/init.d/lirc start
sudo mv /etc/lirc/lircd.conf /etc/lirc/
lircd_original.conf
```

Then copy over your new configuration file:

```
sudo cp ~/lircd.conf /etc/lirc/lircd.conf
```

Your original configuration file will be saved as lircd_original.conf.

16 Restart the LIRC

Now you have a configuration file you can use your remote control. Restart the LIRC by typing **sudo /etc/init.d/lirc restart**. As before, you can test that the lircd file is working by listing all the registered KEYS stored in the file; just type: **irsend LIST the_name_of_your_remote " "**. Send some commands to your television or device using the code:

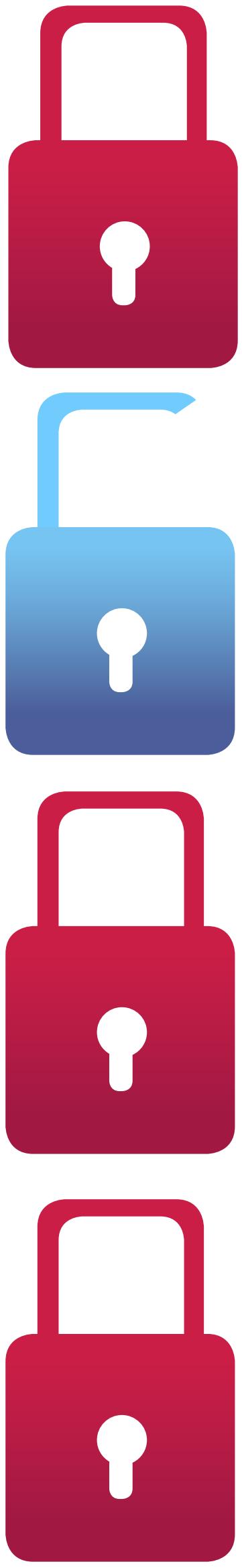
```
irsend SEND_ONCE Remote_Name Remote_Button
```

For example, to control the TV Menu you would type in the LXTerminal: **irsend SEND_ONCE Samsung KEY_VOLUME_UP**. This will send the 'volume up' signal.

17 Common errors and code recap

Transmission error – this usually means that the lircd.conf file is not correct and contains an error. For example, using a file from 2007 instead of the latest version.

Connection refused – generally, this means the LIRC has



failed or the hardware changes are not correct. Check the boot.config file and the hardware.conf file, then restart the LIRC by typing:

```
sudo/etc/init.d/lirc restart
```

Restart the LIRC program – this one can prove useful after you have changed a file, such as the hardware.conf or lirc.conf:

```
sudp/etc/init.d/lirc restart
```

Stop the LIRC program – useful when testing the program:

```
sudo /etc/init.d/lirc stop
```

Start the LIRC program –

```
sudo /etc/init.d/lirc start
```

List all the Keys in the file –

```
irsend LIST Samsung “” #
… replacing Samsung with the name of
your remote.
```

Create a new bespoke lircd configuration file –

```
irrecord -d /dev/lirc0 ~/lircd.conf
```

The Cold Boy

Console modder Daniel d'Entremont won the 2015 BitFix Build-Off with his Zelda-playing mini fridge

The Cold Boy looks fantastic! So, why a fridge? What inspired you?

Well, I was sitting in my chair at home and contemplating whether or not I'd be able to finish a current project that I was working on in time for the contest on BitFix [http://bitfixgaming.boards.net/], the 2015 Game Boy Classic Build-Off. I was scrolling through images online and I somehow found these fridge magnet decals that you could put on your fridge. They did absolutely nothing, but they made it look like a Game Boy. So then I looked over at my fridge and one of my computer monitors was just sitting right there beside it, so I was like, 'Hey, I bet I could make one!' That's more or less how that went.

What was the brief for the contest?

It basically had to be done over a three-month span throughout the summer, and it had to be somehow related to a Game Boy and be able to play some sort of game. So it could have just been a painting of a Game Boy, but one that actually worked, that kind of thing.

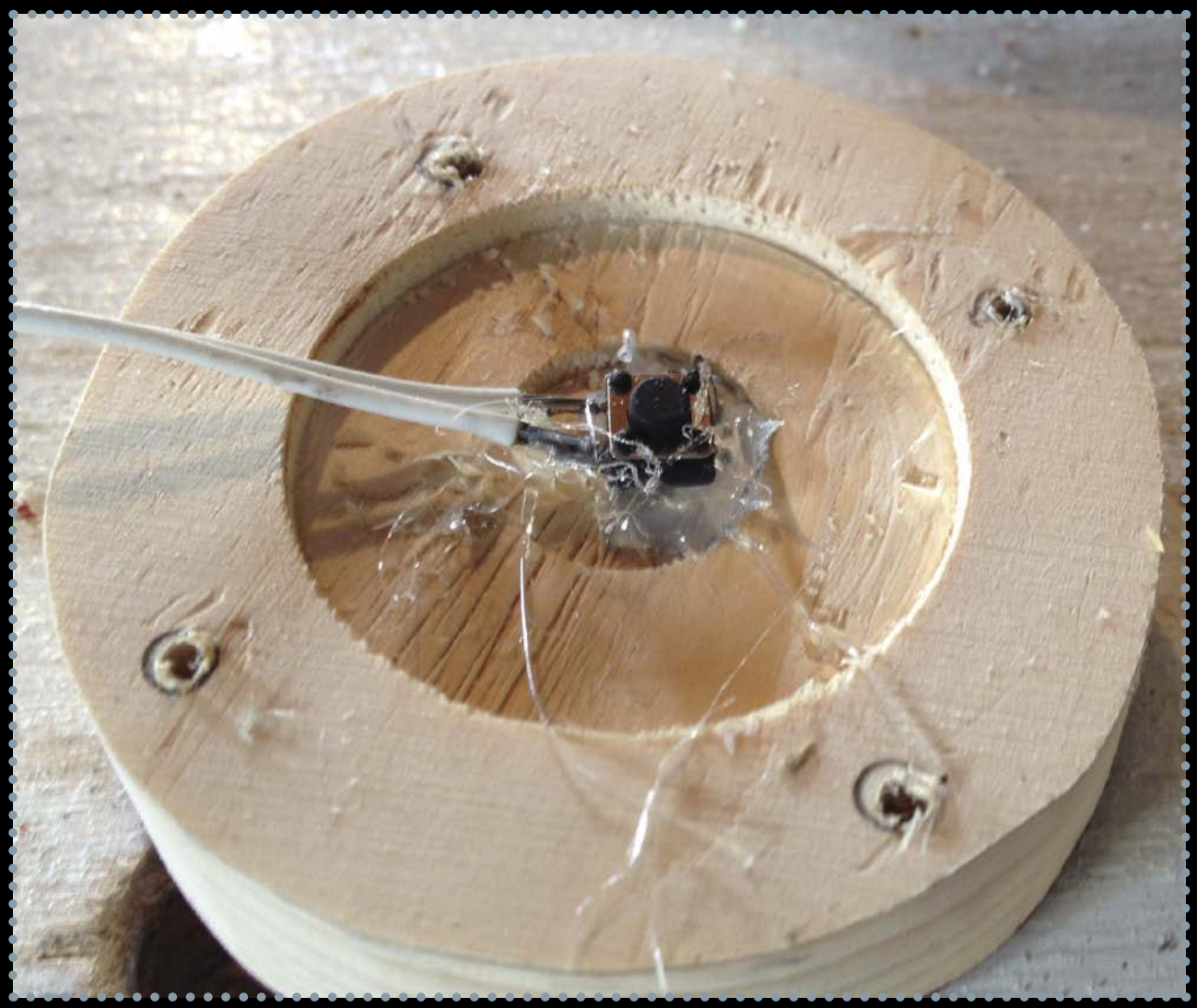
Are you using a regular computer monitor for the display?

Yep, and the bezels are actually just the bezels from the computer monitor as well. It just happened to look right so I kept that. It's just a monitor that I bought at a thrift store called Value Village for about four, five dollars, something like that. It's not the newest thing ever, but it works more than well enough for what I'm doing.

Daniel d'Entremont is a university student and an experienced modder with a passion for console hacks. He does not claim to have perfect mods, but he is a mod purist and strives to keep his mods looking as original as he possibly can

What about the rest of it, how did you create the buttons for the front?

My original plan was that I was going to cut holes in the front in the shape of the buttons and then make the buttons inset in. But due to a lack of time – because this was almost at the end of the contest, which ended on Monday [14th September 2015, by the way, and I did win) – what I did was just made them the shape they are, put them on top of the fridge and put little tiny tact switches in behind them. So I drilled a little hole in the back of each button for the tact switch and inset that button inside the wooden button, then put that on the front. Those are all wired up to a USB keyboard; so the USB cord goes out the bottom of the fridge to the back as well as the video and power. The rest of the fridge is completely normal. I didn't touch the main section of the fridge, I only touched the door.



THE PROJECT ESSENTIALS

Raspberry Pi 2
Fridge
LCD monitor
USB keyboard
Tact switches
Wooden buttons
Insulation

Congratulations on winning the competition! Why did you decide to use a regular keyboard instead of the GPIOs, for example?

I specifically did that because I wanted to be able to hook this up to any computer. I would have done GPIO and I would have stuck the Raspberry Pi right inside of it with the Wi-Fi module had I not been planning to switch the computer out at some point. I wanted to have the freedom to be able to put an old computer inside it, so that way I wouldn't have my Raspberry Pi taken up for this. I don't have umpteen amounts of budget money, so I just figured I could use my Raspberry Pi for the project now.

Is the Raspberry Pi currently doing anything else right now – serving media, for example?

Not right now. I only finished the project a little while

Left Daniel cut the wooden buttons to shape using a band saw and a belt sander, then connected them to tact switches

ago and then I moved to university, so I haven't really had time to do anything. I've been thinking about making it a fileserver, but I haven't had time to do much of anything like that.

If it stays where it is then it will most likely end up doing what it's been doing, but also have a fileserver on it, so then you can just put games on the system by adding the ROM to the fileserver.

What are you running on the Raspberry Pi – is it a RetroPie installation or a distro like Lakka?

It's RetroPie version 3.0. The way RetroPie works, you

Left He used a multimeter to see which trace lead to which key on his USB keyboard, then wired the tact switches to those keys

put a ROM in the ROM folders on the system; they're already preconfigured and all of the emulators are already built in. So when you stick a Game Boy ROM in the Game Boy folder, the emulator turns up on the main menu, but it's already there in the background. So I could run pretty well any other game on there that's eight buttons or just have an external controller for it if it has more than eight buttons.

Do you have any plans for a multiplayer mode? Maybe using more of the USB-keyboard buttons or a web interface for smartphones?

Those are all great ideas, but it's all stuff that will happen in the future. I haven't done anything yet for it because, like I said, I have a lack of time right now, but those are all possibilities. It would be really cool to get hooked up to have some wireless NES controllers; I've already built one wireless NES controller for the PC, so I could hook that up to it and get a few more in, that would be cool. They have the same amount of buttons, right?

Is your fridge still fully functional with all these modifications?

The actual refrigerator parts are fully functional and the door has also been re-insulated. I took the whole door off – you know how you can just take it off and attach it on the other side, and then you can open it up from the other direction? Well, I took it apart enough to do that and then I took apart the actual door. There was a bunch of screws around

the perimeter, so I took those all off and took the inside section of the door off and modified the front half, which had a whole bunch of metal and hard insulation inside it.

When I was re-insulating the fridge, I assumed my Dad would have some insulation somewhere, which he did, but it ended up being ROXUL insulation — a certain type that's fireproof and all this stuff, so I didn't want to use that. What I did was I took a layer off one of his sheds outside – not a whole panel of insulation, but just one flake layer – and I used that. So that was interesting.

Like it?

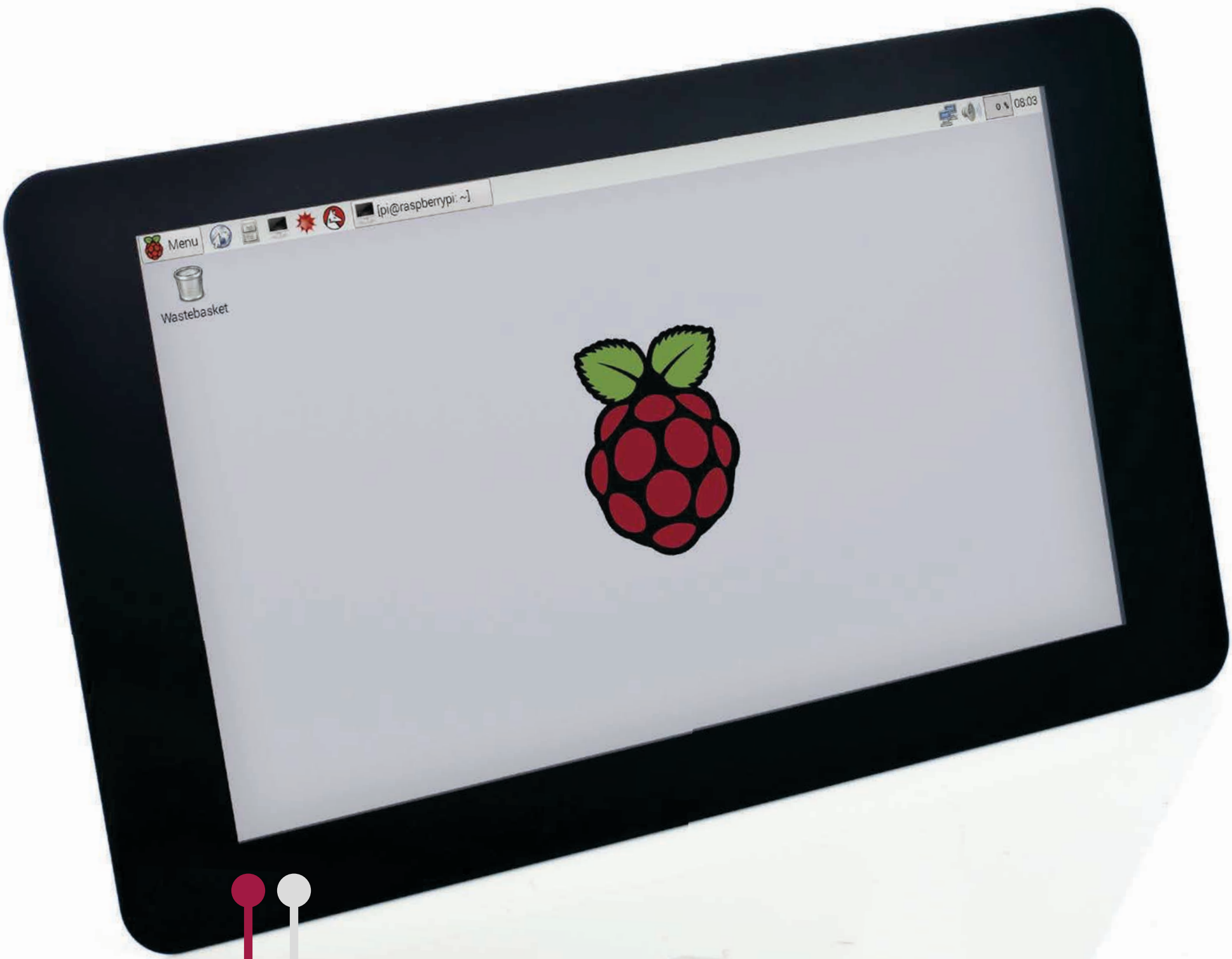
Interested in the other BitFix competition entries? Check out the blog post at http://bit.ly/1KuyrDK to find a SNES/NES laptop mash up and a Portal-themed Companion (Game) Cube. One of the admins even made a Guitar Boy (http://bit.ly/1NODkNV)

Further reading

For more on Daniel's other console hacks, check out his ModPurist website at http://modpurist.ca/. You can also see the wireless NES controller that he mentions in action: http://bit.ly/1Kuzfsf

Set up the official 7-inch Pi Display

Assemble this display module and get it up and running with your Pi



The Raspberry Pi Foundation has had plans for an official display since the very beginning. Development of the display module began in 2013, but as director of engineering Gordon Hollingworth explains in his blog post about it (http://bit.ly/1Z75He0) there were a number of issues along the way, including EMC (electromagnetic compatibility) testing and the sourcing of a high quality yet affordable display (which, from first-hand experience, are not easy hurdles to jump). However, the output device is exactly what you would expect from the Raspberry Pi Foundation: a beautiful 7-inch multi-touch capacitive screen for the incredible price tag of just $60 (plus local taxes and shipping, which in the UK will come to around £51 including delivery). In this tutorial, we will show you how to put it all together and then get up and running with your Raspberry Pi.

THE PROJECT ESSENTIALS

Raspberry Pi official 7-inch display

Power supply

Small Phillips screwdriver

01 Order the parts

You can order the Raspberry Pi display through all of the normal channels. Shop around if you like, but take a look at CPC (**http://pc.farnell.com/SC13858**), RS Components (**http://bit.ly/1OVz8vW**) and the Raspberry Pi Swag store (**http://bit.ly/1QGvp3E**), which all seem to have the best prices available at the moment. However, the stock levels are a bit short because there has been a huge demand for the display, so you may need to look around some of the smaller outlets to find somewhere with it still in stock.

02 Prepare your Pi

As with all Raspberry Pi projects, it is important to make sure your device has the most up-to-date version of any

software you will be using. For this guide it is even more important as the drivers for the display have only recently been released (they don't even feature in the latest version of NOOBS). Open a terminal window and type:

```
sudo apt-get update
sudo apt-get upgrade
```

Once this has completed, shut down your RasPi using the command **sudo shutdown –h now**.

03 Unpack the screen

The screen comes as a kit of parts, as can be seen here. Included is the display itself, a driver board, a ribbon cable, mounting hardware and jumper cables. The packaging box should also include some pink anti-static foam and it is recommended to keep the flat part of this to use as a work surface, to avoid scratching the screen. For the time being you should also leave the protective film on the front of the screen, and possibly keep it on until it is installed in its final location.

04 Attach the driver board – part 1

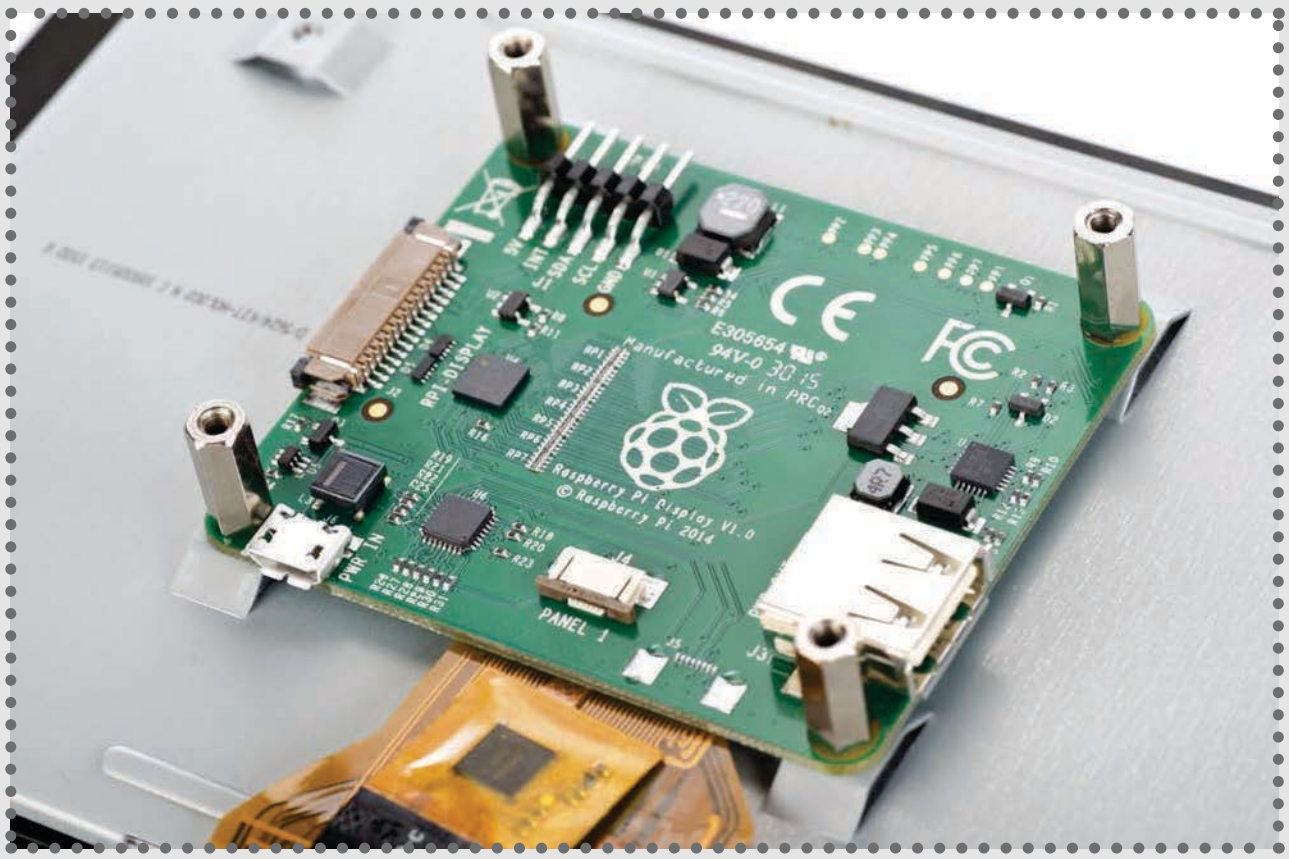
Lay the display face-down on the anti-static foam, and you will see on the back that there is a large ribbon cable with a smaller ribbon cable attached. Grab the driver board, where on the back there should be a large connector. You should also lay this face-down on the anti-static foam.

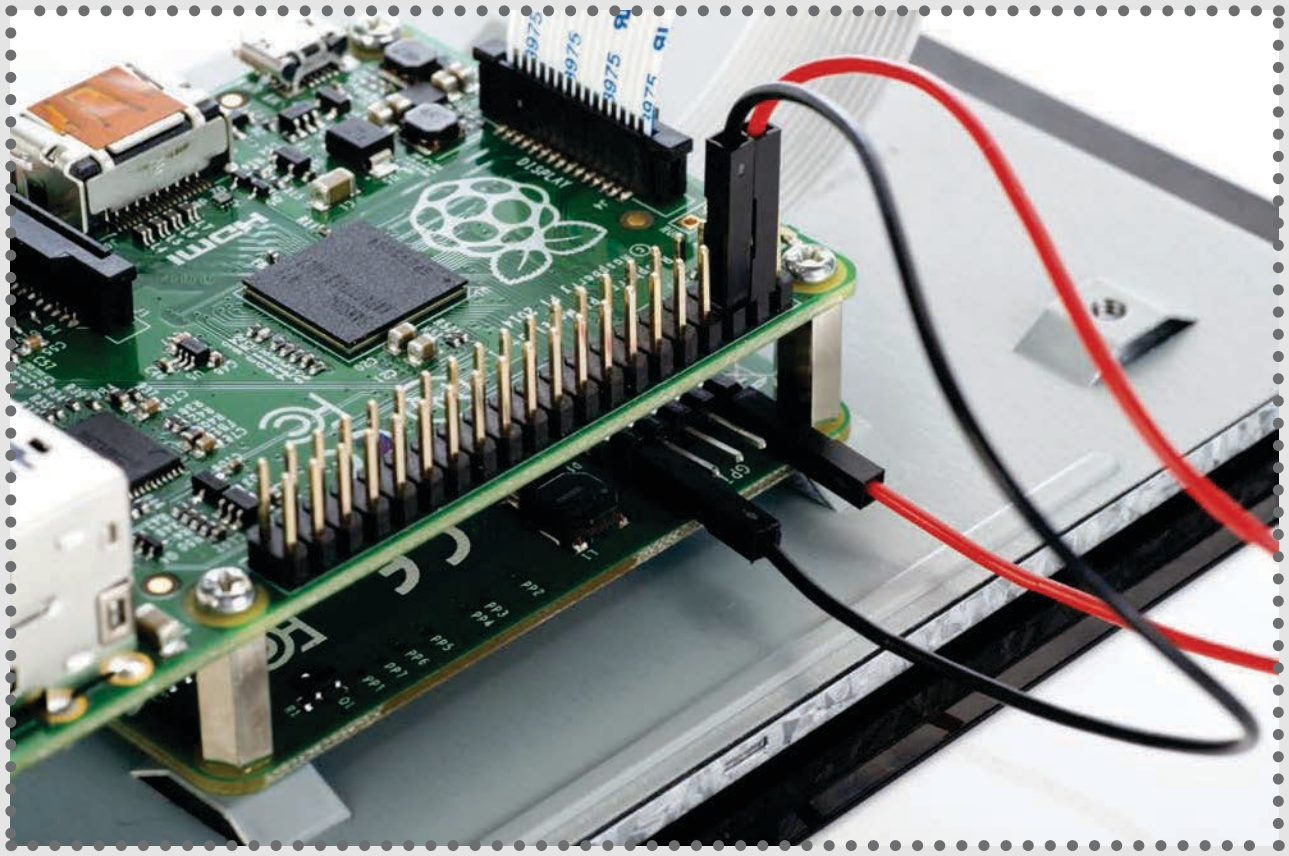
05 Attach the driver board – part 2

Carefully unclip the large connector on the driver board, insert the large ribbon cable and then close the clip again to secure the cable. Turn the driver board over onto the back of the display, where you should now see a small ribbon cable and a small connector on the top side of the board. Connect this smaller ribbon cable in the same way you did the large one.

06 Secure the driver board

Once the ribbon cables are securely fastened, the next step is to secure the driver board to the back of the display. Grab the mounting posts, align the four holes on





the board with the holes on the display and then screw the driver board into place with the mounting posts. Be careful not to overtighten.

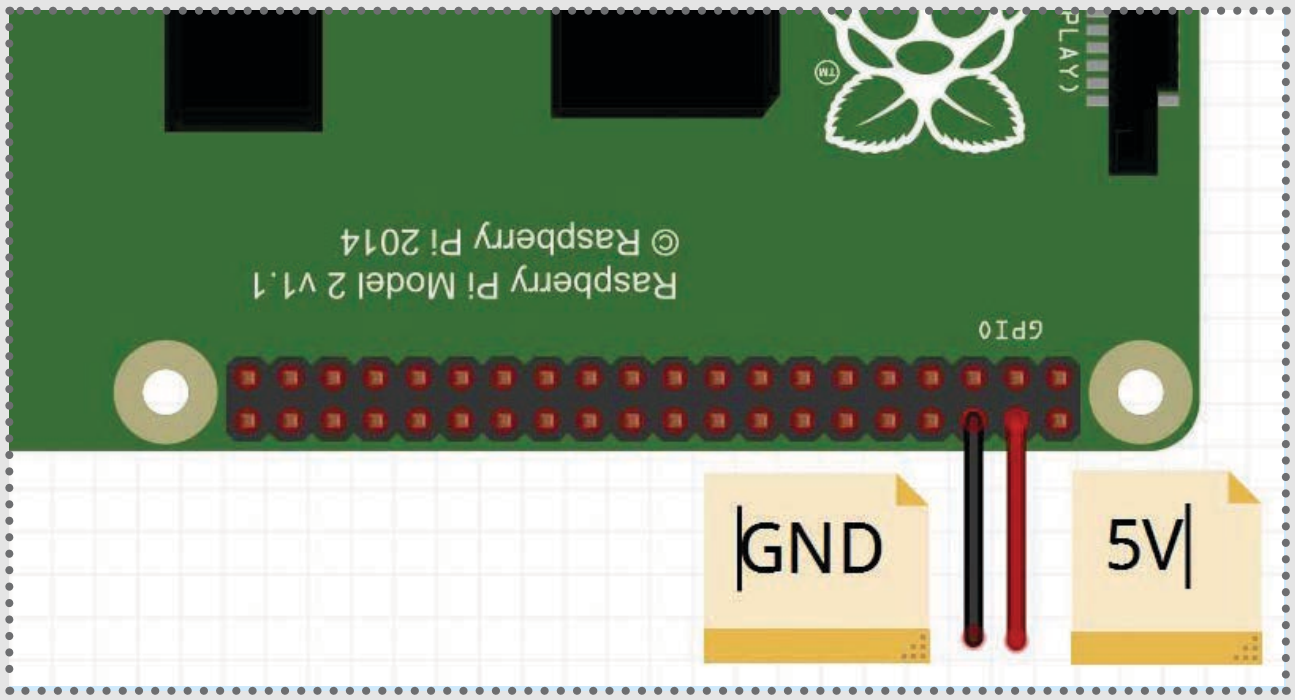
07 Attach the Raspberry Pi – part 1

On the top of the driver board, on the opposite side to the USB connector, you should see another connector. Grab the white ribbon cable that came with the kit and connect it here with the blue mark facing the back of the screen. Next to this, you should see another connector with five pins. Plug the red jumper cable into the one marked '5V', then plug the black jumper cable into the one marked 'GND' (the colour of each cable doesn't matter, but this is the convention).

08 Attach the Raspberry Pi – part 2

Place your Raspberry Pi on top of the mounting posts with the USB port on the same side as the USB ports on the display driver board. Screw the Raspberry Pi into place with the provided screws; once again, be careful not to overtighten. Then you can attach the other end

of the white ribbon cable into the DSI connector on the Raspberry Pi, which is the one on the same side as the ribbon cable that is already attached. It should loop round easily and plug in.



09 Power your Pi

The display has been designed so that the driver board and Raspberry Pi can run from a single power supply. Take the red jumper cable (or whatever colour you used for 5V) that you plugged into the driver board in Step 7 and connect it to pin four on the Pi GPIO (second down, closest to edge of the RasPi board). Then take the black or GND jumper cable and connect it to pin six on the RasPi GPIO (third one down, closest to edge).

10 Boot your Pi

You should now be able to plug in your 2A micro-USB power supply into the connector on the display driver board, then the Raspberry Pi and display will both boot up. The touch functionality works through the white ribbon cable, so no other connections are necessary and it should work straight away!

Customise the display

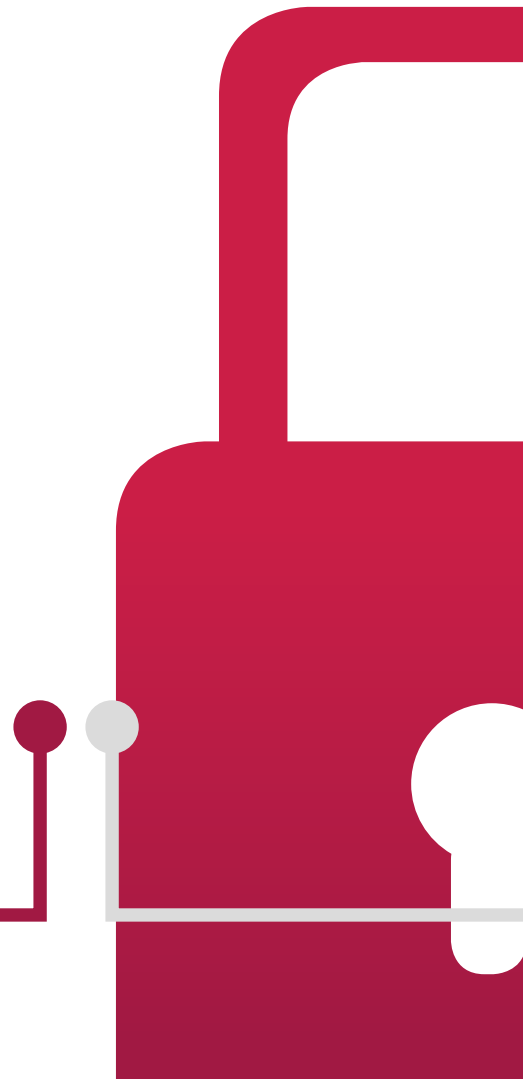
Even though the display has only been available for a relatively short period of time, a number of companies, such as Pimoroni and ModMyPi, have created cases for the Raspberry Pi Display to enable you to customise and protect your new device. They are all reasonably priced at around £10, and come in a variety of colours and styles to suit your needs.

Set up a WordPress website on your Pi

Here's a quick guide to running a WordPress server and setting up your Raspberry Pi as a LAMP Stack

One of the main reasons that the Raspberry Pi has been so popular in recent years is because it is an extremely capable little computer that packs a lot of punch for its very small price tag. Since the release of the quad-core, BCM2836-based Raspberry Pi 2 Model B in the early months of 2015, it is now even more capable than ever before and makes the perfect platform for your very own low-powered, micro-sized home web server.

In this tutorial we will show you how to set up a LAMP stack (Linux, Apache, MySQL and PHP) on your Raspberry Pi and then install a WordPress site on top of this. WordPress is one of the most popular content management and blogging platforms on the web, with over 52 million new posts uploaded each month. According to the company's own statistics (**https://wordpress.com/activity**), there are over 409 million people viewing more than 19 billion WordPress pages each month. So as far as website software goes, WordPress is one of the best out there that is available to you – so let's get set up.



01 Update your Pi

As with all Raspberry Pi projects, it is best to update your Pi to the latest OS firmware before proceeding. You can do this by opening a terminal session and issuing the following commands:

```
sudo apt-get update
sudo apt-get upgrade
```

When prompted, press Y and then Enter to confirm.

02 Install the necessary software

As mentioned in the introduction, a LAMP stack uses Linux (the operating system), Apache (the web server software), MySQL (the database software) and PHP (the programming language for web). We have the Linux part covered, but we will need to install the rest. To do this, open a terminal window and issue the following:

```
sudo apt-get install apache2 php5 libapache2-
    mod-php5 mysql-server php5-mysql -y
```

When the process starts installing MySQL you will be prompted to enter a root password. Make sure you remember this password as we will need it later on to enable your website's access to the database.

03 Test Apache

You can test the installation of Apache by opening a web browser on your Pi and browsing to **http://localhost/** – that should take you to the default Apache 'It works!' page.

Using a pretty web domain

If this is the first time you have set up a web server, you are probably now thinking about how you can access the website from outside of your home network and with a domain name of your choice. Unfortunately, we do not have space to cover that here, but you can easily do this with a dynamic DNS service and a custom domain of your choosing. Dig around on Google and you will find plenty of advice.

04 Test PHP

Open a terminal window and then browse to the web server root with:

```
cd /var/www/
```

Next, delete the default index.html page and then create and open an index.php page for editing by entering:

```
sudo rm index.html
sudo nano index.php
```

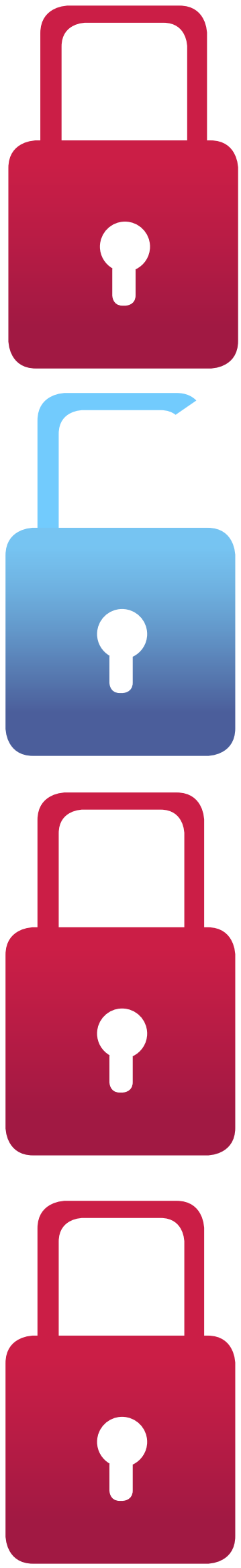
Inside the index.php file, enter the following line and save the file:

```
<?php echo date('Y-m-d H:i:s');
```

Go back to your browser, where you tested Apache, and refresh the page. You should now see a dynamic time and date display.

05 Download WordPress

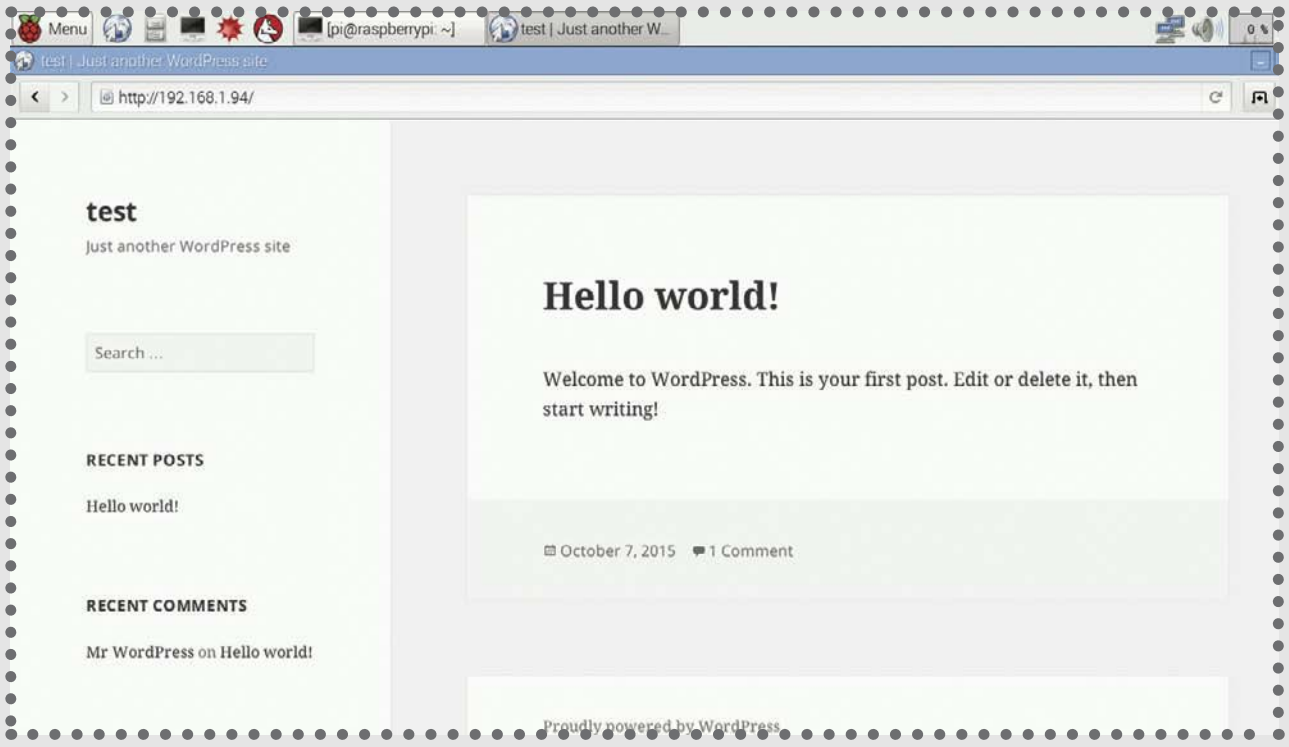
To download WordPress you need to open a terminal window and navigate again to the web server root. Then change the ownership of the folder to the 'pi' user, delete all existing files located there and download the WordPress files. To do this, issue the following commands in order:



```
cd /var/www
sudo chown pi: .
sudo rm *
wget http://wordpress.org/latest.tar.gz
```

The final step is then to extract the tarball, move all the files to the web server root and then go back to remove the original download. Do that with:

```
tar xzf latest.tar.gz
mv wordpress/* .
rm -rf wordpress latest.tar.gz
```



Left Once everything is correctly set up, you'll be greeted with this intro page

06 Set up the database

Run MySQL and log in using the following command:

```
mysql -uroot -ppassword
```

...where 'password' is the password you set in Step 2. There is no space between –u and root or –p and your password. Once you are logged in to MySQL, create the database using:

```
create database wordpress;
```

The trailing semicolon is important and the command will not complete without it. You can then exit MySQL and go back to the terminal's command line using Ctrl+D.

07 Configure WordPress

First you need to find out your Rasberry Pi's IP address on your local network by typing:

```
hostname -I
```

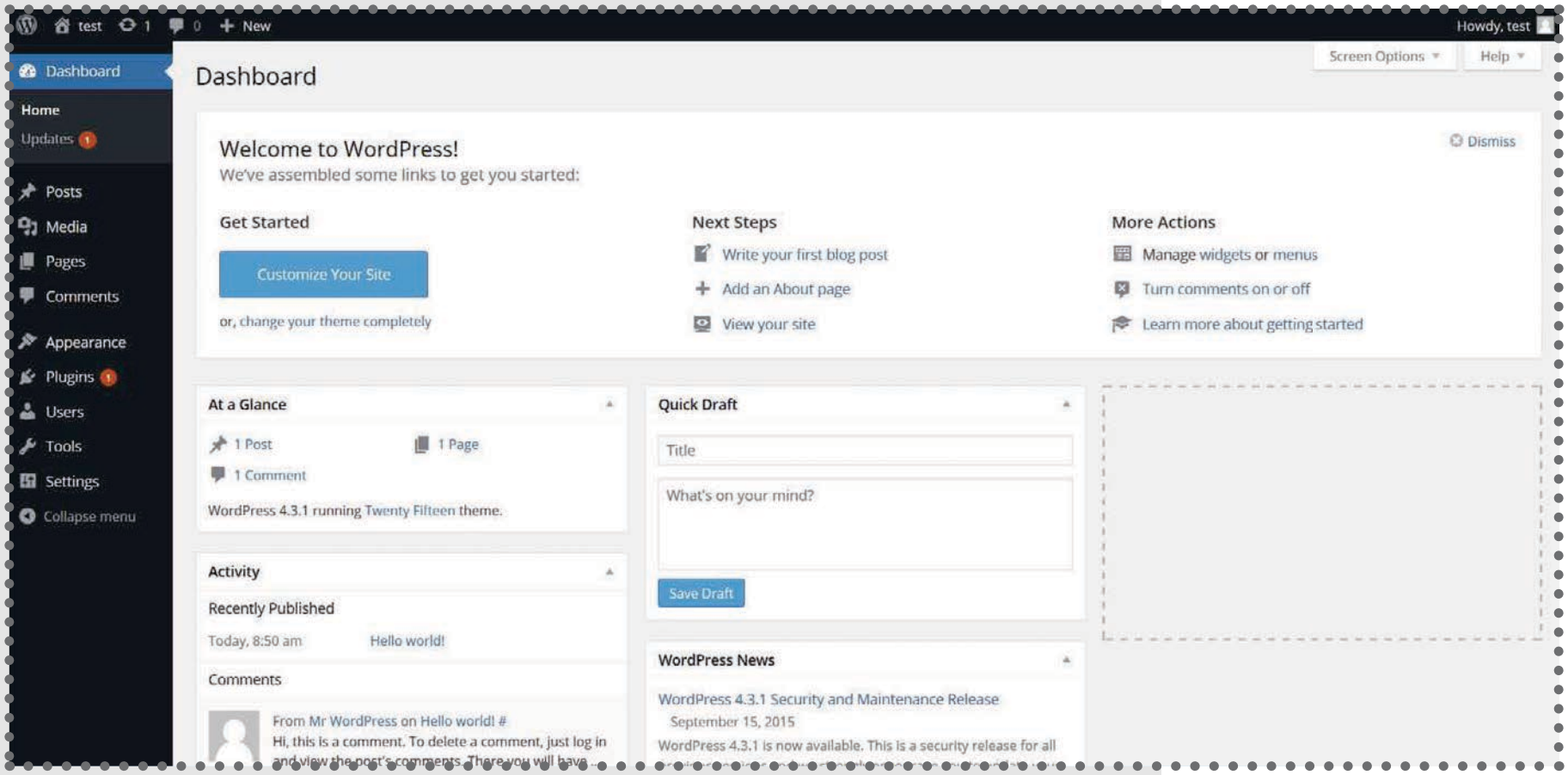
This should display your IP address. Open a browser and navigate to **http://192.168.xxx.xxx**, replacing the 192.168.xxx.xxx at the end with your own IP address. On the WordPress error page, click 'Create a configuration file' and on the next page click 'Let's go'. You should then fill in the basic database information with the details we set up previously:

```
Database Name -  wordpress
Username -       root
Password -       your-password
Database host -  localhost
Table prefix -       wp_
```

Once the script successfully connects, you should see a new page containing some PHP code.

08 Install WordPress

Copy the text that loaded into the page on Step 7. Minimise the web browser and go back to the terminal session. Now create a wp-config.php file by typing:




```
nano wp-config.php
```

Paste the copied text into the file, press Ctrl+X, Y and then Enter to save. Go back to the web browser and press 'Run the install'. You should now be greeted by a Welcome page where you can enter your site title, username, password and some other details. Once complete, you can then visit **http://192.168.xxx.xxx/wp-admin** (replacing 192.168.xxx.xxx with your actual IP address) from any computer on your local network and you will be taken to your WordPress dashboard.

09 All done

You now have a fully functional WordPress website running from your Raspberry Pi. Log in to the dashboard, add some pages, posts and customise it to your heart's delight!

Turn your Raspberry Pi into a stop-motion studio

Build your own animation studio by using your Raspberry Pi as a stop-motion camera

What have you done with your Raspberry Pi camera lately? While it gives us plenty of new ways to use the Pi, unless you've got your computer set up as a security webcam or you're a particular fan of time-lapse photography, the chances are that you've overlooked the Pi camera module for a while.

If you're a fan of animation or you simply want to extend the possibilities of the module, why not build a stop-motion camera? By using Python and an external button to capture images, the Raspberry Pi can be the perfect tool for animators.

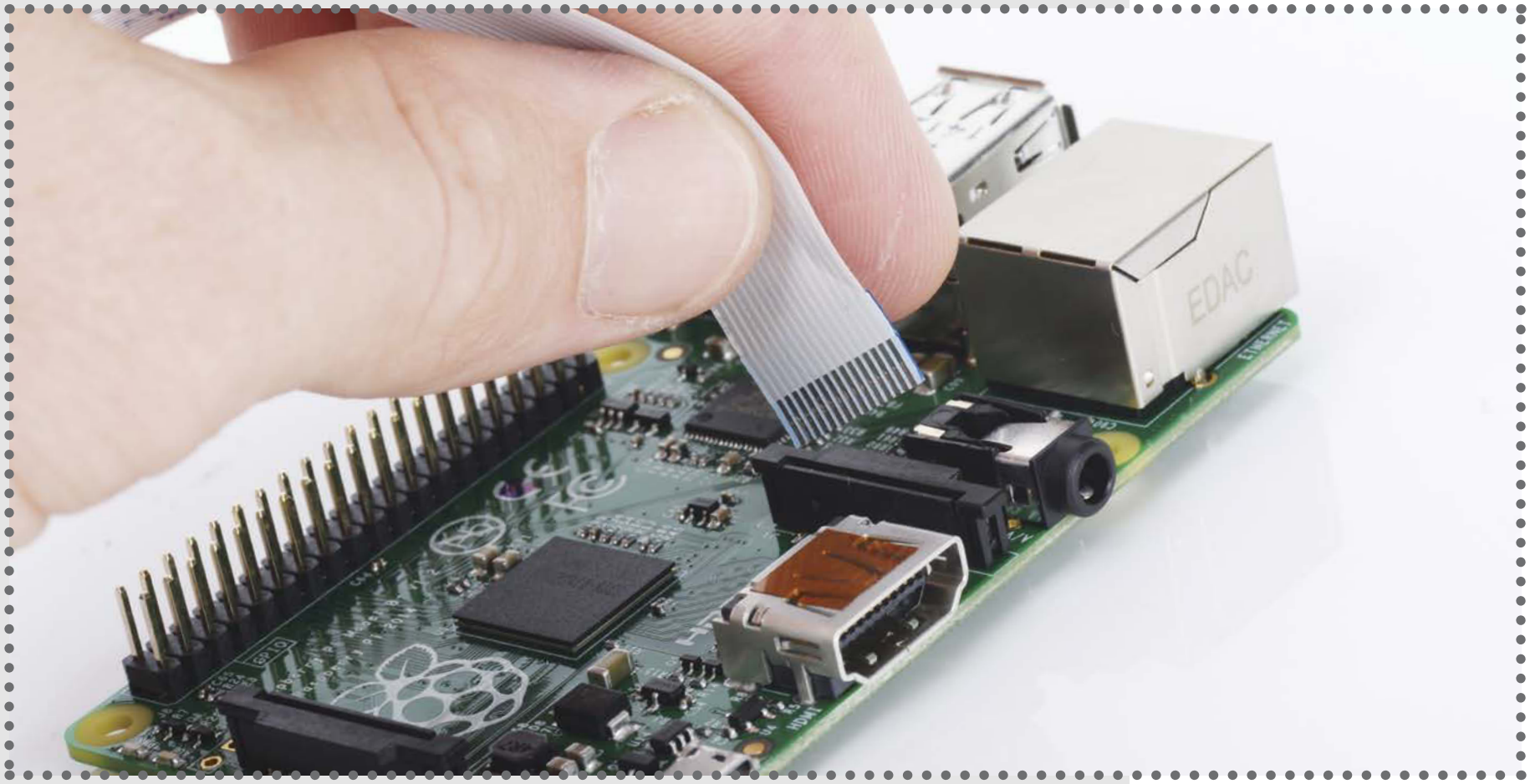
Better still, you can go beyond animating toys or bits of LEGO and go old school by mounting the Pi on a rostrum and creating a cartoon. Even if you can't buy or build one, you can mount the stop motion Pi camera with a smartphone mount for stability.

01 Mount your stop-motion Pi camera

Before you get started, think about the type of animation you're going to be capturing. If you're using the traditional top-down method, as used by classic cartoon animators, then you'll need a rostrum to mount the Raspberry Pi.

Alternatively, you may be animating something on a desk, table or perhaps the floor, but you'll need your Pi camera mounted in a similar way, looking across rather than down.

Various options are available, such as smartphone tripods and dashboard mounts. Most of these should be suitable for securely mounting your Raspberry Pi.



Above With the camera module, ensure the shiny side faces away from the Ethernet port

02 Find somewhere to shoot

For your first attempts at shooting a stop-motion video, you should use a wide and uncluttered space. This might be a desk, a kitchen work surface or even the floor, but it should be a hard and flat area in most cases (unless you have need for a bumpy carpeted environment for your video) to aid with the creation of your stop-motion film.

As time progresses and your skill develops, other surfaces can prove useful alternatives, but keep it simple for now and stick with flat surfaces while you get to grips with the art form using the Raspberry Pi stop-motion camera.

03 Connect the Pi camera module

Next you'll need to connect the Pi camera module to your Raspberry Pi. All models have the necessary connector, although where it is found on the device will depend on the version of your Raspberry Pi.

The Model A has the Pi-camera connector next to the Ethernet port, as does the Model B. On the B+ and the Raspberry Pi 2, the connector is in a similar position, but it's a little further from the Ethernet port between the audio-out and HDMI ports.

Connecting the camera module can be tricky. Begin with taking your Pi out of its case or remove the top where possible and disconnect all cables. Take precautions before removing the device from its antistatic bag, as the camera module is very sensitive to static electricity.

On the Pi, lift the plastic catch on the connector and slot the camera module flex into place with the shiny contacts facing away from the Ethernet port. Once the flex is fully slotted in, push the plastic catch back into place.

04 Test your Pi camera module

After connecting the Pi camera, check that it works by booting the Raspberry Pi (we're assuming you're running Raspbian) and entering this in the command line:

```
sudo raspi-config
```

With the keyboard arrows, move down to option five, 'Enable Camera', and tap Enter. In the following screen, hit Enter again to enable the camera and exit.

If you're not already signed into the GUI, do so now (if you're in the command line interface, enter **startx** to launch the desktop view). Open the terminal and enter:

```
raspistill -o image1.jpg
```

You can review the resulting image in your Home directory.

05 Straighten out the image

With the Pi camera up and running, you may notice that it's outputting the image with the axes flipped. We can fix this using Python, so open the terminal and enter:

```
sudo apt-get install python-picamera python3-picamera
sudo idle3
```

In the Python editor, open File>New Window and enter the code below, setting the camera.vflip and camera.hflip as True or False as required. Save (perhaps as 'camflip.py'), then press F5 to run the script and view the correctly outputted image.

To save time, however, you might try rotating the position of your camera or Pi camera module!

```
import picamera
from time import sleep

with picamera.PiCamera() as camera:
    camera.vflip = True
    camera.hflip = True
    camera.start_preview()
    sleep(3)
    camera.capture('/home/pi/image2.jpg')
    camera.stop_preview()
```

"You may notice that it's outputting the image with the axes flipped. We can fix this using Python"

06 Set up the breadboard and button

We have two ways to add a button to the Raspberry Pi, but before proceeding, ensure you have switched the computer off and disconnected it from the mains. You should also disconnect any cables and hardware.

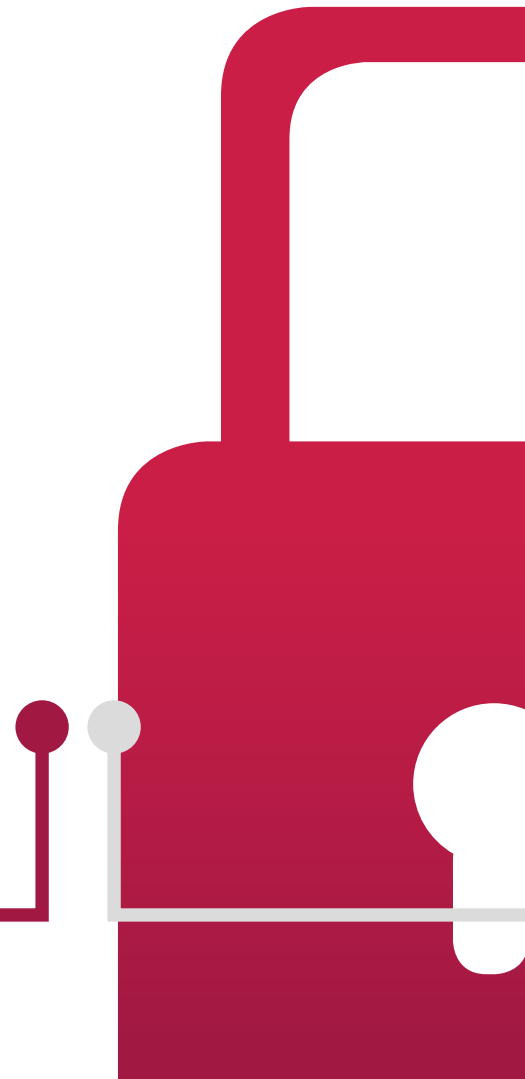
The simplest method of adding a button is to employ a solder-free breadboard and a single-state pushbutton. Connect the button to the breadboard with two male-to-female wires running to GPIO pins GND and 17. With a script designed to detect action from the button on the GPIO, each frame of your animation can be captured with a single button push.

Learn about your Pi

Look at how you can monitor the Raspberry Pi itself to see what is happening

In previous articles, we have looked at how you can use your Raspberry Pi in order to monitor the environment around itself. However, we haven't looked at how the Raspberry Pi can monitor and see what is happening within itself. Luckily, Broadcom has included a command line utility, named vcgencmd, that is designed to deliver to you exactly that type of information. You can get a full list of the commands available with the command vcgencmd commands. Currently, the most complete documentation is available from the website **http://elinux.org/RPI_vcgencmd_usage**, but there is no official documentation available at the time of writing this article.

The first step is to take a look at using the vcgencmd directly to see what you can do with it. We will then look at how to incorporate this into a Python program. The first piece of information you may want is the firmware version for your Raspberry Pi. You can get this with the command **vcgencmd version**. One of the issues you may come upon is the way that



memory is handled on the Raspberry Pi. The pool of RAM is divided between that available to the main CPU and that available to the GPU. You can check to see what the split is with the commands **vcgencmd get_mem arm** for the CPU portion and **vcgencmd get_mem gpu** for the GPU portion. For people who are overclocking their Raspberry Pis, there are a few commands that are essential. The first is the ability to check the relevant frequencies and verify that they are what they should be. The command **vcgencmd measure_clock <clock>** will return the actual frequencies for various clocks used within the Raspberry Pi. The core clock frequency can be measured with the command **vcgencmd measure_clock core**. You can also get the CPU clock frequency by using the 'ARM' clock. On this author's Raspberry Pi, there are 12 separate clocks that can be measured. When you do overclock your Raspberry Pi, you need to be very careful of generating excess heat. Luckily, you can measure the internal temperature with the command **vcgencmd measure_temp**. This returns the internal temperature of the BCM2835 SoC in degrees centigrade. For any American readers out there, you will need to do a small conversion in your code if you want that temperature in degrees fahrenheit. The last item that is of importance to overclockers is the various voltages for the CPU and RAM systems. You can use the command **vcgencmd measure_volts core** to get the core voltage. The various memory voltages can be measured with the options **sdram_c**, **sdram_i** and **sdram_p**. While these commands measure the actual values, you may need to see what the values are supposed to be based on: the configuration

values you set. You can see these with **vcgencmd get_config**. This enables you to make comparisons between what was supposed to be set and what is actually happening.

Now that we can make these measurements, how can you use them within your own program? You need to be able to spawn off a process to run the command in question and then collect the resultant output so that you can do something interesting with it. The first step is to actually run the commands. There are a couple of common ways to get this functionality. The first is to use the subprocess Python module to launch a subprocess and then collect the returned output. You can use the function **check_output()** to launch your vcgencmd and store the results into a variable. You give your command and its command line options to the function with a list of strings, one element for each word. For example, if you wanted to find the current temperature, you could use:

```
import subprocess
temp1 = subprocess.check_output(['/        opt/
    vc/bin
vcgencmd',          'measure_temp'])
    temp2 = temp1.split('=')[1]
    curr_temp = float(temp2[:-3])
```

The second method available to run external commands is to use the OS Python module. In this case, you can use the **popen()** function to launch a subprocess and create a pipe between it and your current Python process. This pipe stream behaves like any other file stream. Consequently, you can use

Why Python?

It's the official language of the Raspberry Pi. Read the docs at https://python.org/doc

the associated **read()** function to get the results back from the vcgencmd command. As an example, say you wanted to get the current CPU frequency that has been configured, you could use something similar to the following:

```
import os
c1 = os.popen('/opt/vc/bin/vcgencmd get_
    config arm_freq')
cpu_freq = c1.read()
```

In both this example and the previous one, we used the full path to the executable vcgencmd. The reason for this is that the value of the PATH environment variable can be different from what you see when you SSH in to your Raspberry Pi, depending on how these Python scripts are run. Whenever you have any doubt as to what the environment variables are going to be set at, it's best to err on the side of caution and use the full path to necessary files, such as executables.

This method of using **os.popen()** to launch a subprocess has been deprecated, which means it will go away sometime in the future. However, there is still lots of code out there that uses it, so it is important to include how to use it here. More importantly, we should also include how you can migrate away from using it with minimal discomfort for you, the program developer. In this case, you can find similar behaviour with the class Popen from the subprocess module. It accepts similar options to the os.popen(), but behaves much better in many more situations. The Popen equivalent to the above example would look like:

```
import subprocess
c1 = Popen('/opt/vc/bin/vcgencmd        get_
    config arm_freq',          stdout=PIPE)
c2 = c1.stdout
cpu_freq = c2.read()
```

As you can see, there are only a few major changes. The first is making a couple of alterations: one import statement and changing os.popen to Popen. The next change is to explicitly turn the standard output stream into a pipe and the addition of the line where we select the standard output stream explicitly in the fourth line.

Since we have introduced Popen in order to be able to use the vcgencmd command line utility, we should see what else we can do with it in some other projects.

These subprocesses created by the Popen object are true subprocesses in the operating system sense. This means that they are running parallel to the current Python process. You can use this to do very rudimentary parallel processing. You can also use this to run some other program in the background by calling **Popen("my_program")** and letting it run for some time. You can then check in on its progress a few different ways. You can use **poll()** to see whether the backgrounded program has finished yet or not. When it has finished, the poll() function will return the exit code from your secondary program. If you can't do anything else until this backgrounded process has finished, you can call **wait()** instead. This function will block until the subprocess has finished its work. If you need to interact with your

"The first step is to take a look at using the vcgencmd directly to see what you can do with it"

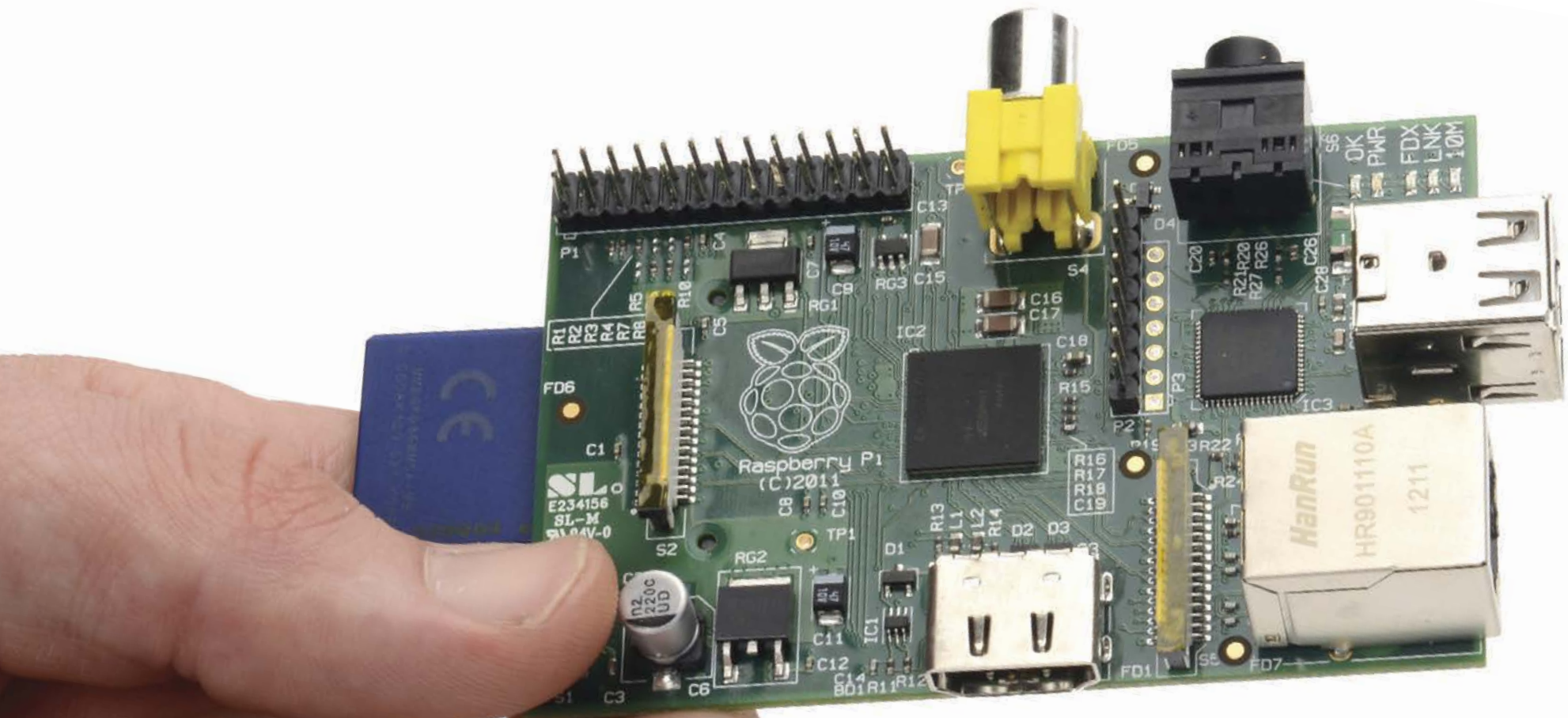
backgrounded process with a bit more functionality, you can use the function **communicate()**. You can give it an option of a string that gets fed into the standard input stream of the background process. You will be returned a tuple containing the data from the standard output and standard error streams. In order for this communication to work, you need to be sure to use the options **stdin=PIPE**, **stdout=PIPE** and/or **stderr=PIPE**. This function blocks on the return until either the background process dies or it gets an end-of-file marker. If you want to be able to interact without having your Python process block, you can write your background program to listen to and act on POSIX signals. You can then send in signals with the function **send_signal()** and have it do interesting things based on these signals. The last thing you might need to do is shutdown this process at some time. You can send a SIGTERM signal to the background process with the **terminate()** function. However, SIGTERM signals can sometimes be ignored. So you might need to send in a SIGKILL signal with the function **kill()**.

Now that you have seen the basics of vcgencmd, you can add the ability of monitoring your Raspberry Pi to you Python programs. This way, you can keep an eye on how your Raspberry Pi is doing when you deploy it into your project.

Talking Pi

Join the conversation at...

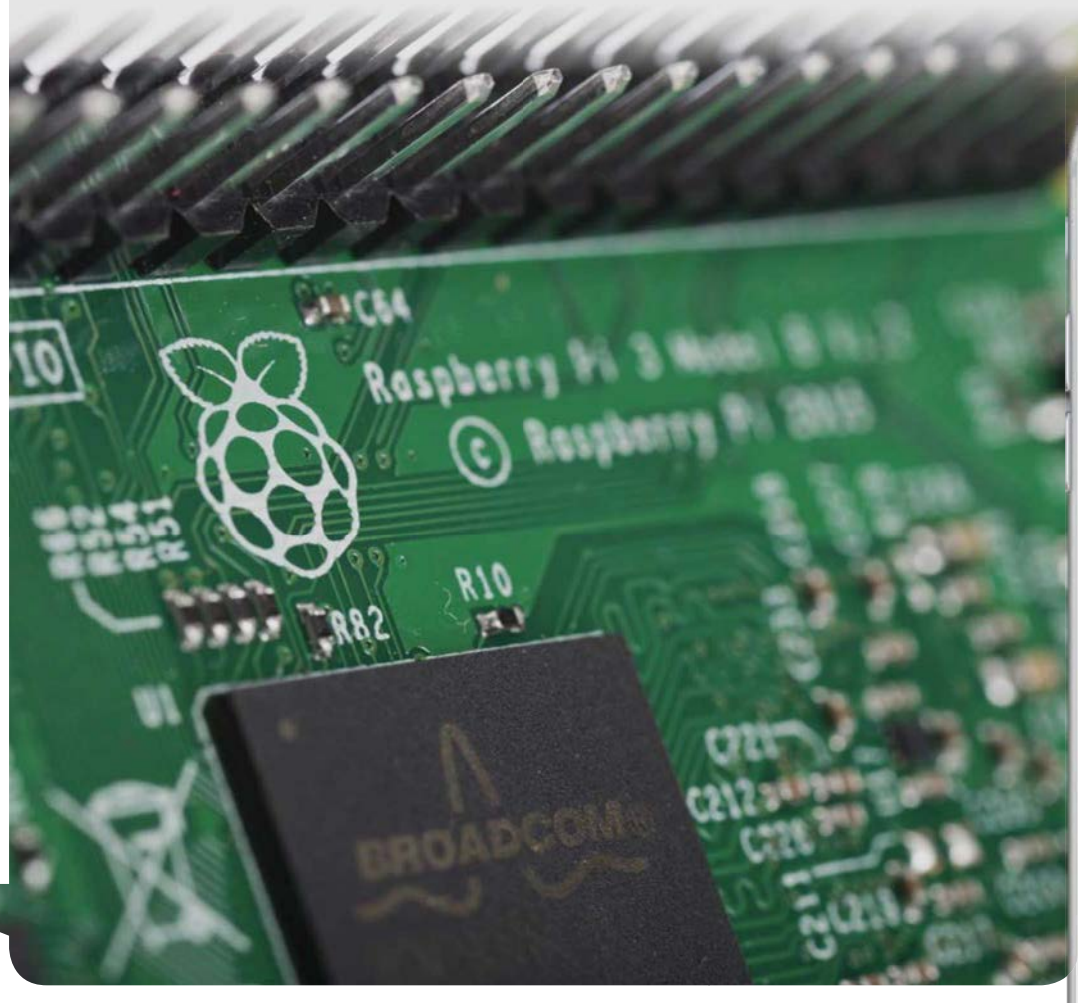
@linuxusermag | Linux User & Developer | @ RasPi@imagine-publishing.co.uk



The Raspberry Pi encourages a lot of hands-on work and this means it doesn't quite work like – or as easily as – your laser focus-tested smartphone interface, or even a normal computer. So we're answering your burning Raspberry Pi questions each issue – get in touch with us on Twitter, Facebook or by email.

Can you recommend any smaller-form alternatives to the Raspberry Pi for an Internet of Things project?
Piers via email

Hello Piers. There are lots of good alternatives to the Raspberry Pi out there, and some of them are designed specifically for Internet of Things projects. Onion (**https://onion.io/**) makes some great ones in the shape of the Omega and the upcoming Omega2. The first board, the Omega, is just 28.2mm x 42mm in size but despite that manages to pack in a 400MHz processor, 64MB RAM, 16MB flash storage, Ethernet, USB and Wi-Fi connectivity and 18 GPIO pins. It costs just $19.99. Its successor, however, is even smaller, even more powerful, can run a full Linux distro and it's fantastic value at just $5 for the board!



Keep up with the latest Raspberry Pi news by following @LinuxUserMag on Twitter. Search for the hashtag #RasPiMag

JUST A SCORE
WHAT'S YOUR JUST A SCORE?

Have you heard of Just A Score? It's a new, completely free app that gives you all the latest review scores. You can score anything in the world, like and share scores, follow scorers for your favourite topics and much more. And it's really good fun!

Is it possible to make a wildlife camera if I buy a Raspberry Pi?

Faith via email

It is indeed, Faith! And the best thing is, you can buy a kit that comes complete with the Pi, camera, weather-proof housing and all of the other equipment you'll need to make and set up your wildlife camera. Visit **https://shop.naturebytes.org/product/naturebytes-wildlife-cam-kit/** to check it out. It's even got infrared to capture nocturnal animals. If your Pi can see your Wi-Fi network from where you've placed it, you can even get it to upload or tweet photos for you.

What happens to Astro Pi now Tim Peake is back from the ISS?

Oscar via email

Tim Peake may be back on Earth, but his Raspberry Pi colleagues Ed and Izzy's continuing mission will last until 2022 (that's when it's thought that their batteries will finally run out). The next person to take charge of Astro Pi projects aboard the International Space Station will be French astronaut Thomas Pesquet, who goes up in November 2016, followed in May 2017 by Italian astronaut Paolo Nespoli. There will be plenty of new Pi-based projects to keep both astronauts busy, as the European Space Agency is sending out even more Astro Pi kits to schools in all 22 member states of the ESA.

Next issue

Get inspired **Expert advice** **Easy-to-follow guides**

Hack a robot with Pi

Get this issue's source code at:
www.linuxuser.co.uk/raspicode